

THE LOGIC OF THE REVERSE MATHEMATICS ZOO

GIOVANNA D’AGOSTINO AND ALBERTO MARCONE

ABSTRACT. Building on previous work by Mummert, Saadaoui and Sovine ([MSS15]), we study the logic underlying the web of implications and nonimplications which constitute the so called reverse mathematics zoo. We introduce a tableaux system for this logic and natural deduction systems for important fragments of the language.

CONTENTS

1. Introduction	1
2. Basic observations about s-logic	2
3. Tableaux for s-logic	6
4. Natural deductions for fragments of s-logic	7
4.1. A Natural Deduction Calculus for \mathcal{F}_2	7
4.2. A Natural Deduction Calculus for \mathcal{F}_1	9
4.3. A Natural Deduction Calculus for \mathcal{F}_3	10
5. \mathcal{F}_2 and Prolog	14
References	15

1. INTRODUCTION

Reverse mathematics is a wide ranging research program in the foundations of mathematics: its goal is to systematically compare the strength of mathematical theorems by establishing equivalences, implications and nonimplications over a weak base theory. Currently, reverse mathematics is carried out mostly in the context of subsystems of second-order arithmetic and very often a specific system known as RCA_0 is used as the base theory.

The earlier reverse mathematics research, leading to Steve Simpson’s fundamental monograph [Sim09], highlighted the fact that most mathematical theorems formalizable in second order arithmetic were in fact either provable within RCA_0 or equivalent to one of four other specific subsystems, linearly ordered in terms of provability strength. This is summarized by the *Big Five* terminology coined by Antonio Montalbán in [Mon11]. However in recent years there has been a change in the reverse mathematics main focus: following Seetapun’s breakthrough result that Ramsey theorem for pairs is not equivalent to any of the Big Five systems, a plethora of statements, mostly in countable combinatorics, have been shown to form a rich and complex web of implications and nonimplications. The first paper featuring complex and non-linear diagrams representing statements of second order arithmetics appears to be [HS07] (notice that the diagrams appearing in [CMS04, Mar07]

Date: December 22, 2015.

Marcone’s research was supported by PRIN 2012 Grant “Logica, Modelli e Insiemi”.

D’Agostino’s research was supported by the GNCS-INDAM Project “Algoritmica per il model checking e la sintesi di sistemi safety-critica”.

are of a different sort, as they deal with properties of mathematical objects, rather than with mathematical statements). Nowadays diagrams of this kind are a common feature of reverse mathematics papers. This is called the zoo of reverse mathematics, a terminology coined by Damir Dzhafarov when he designed “a program to help organize relations among various mathematical principles, particularly those that fail to be equivalent to any of the big five subsystems of second-order arithmetic”. This program is available at [Dzh]. Ludovic Patey’s web site features a manually maintained zoo ([Pat]). The recent monograph [Hir15], devoted to a small portion of the zoo, features a whole chapter of diagrams. These diagrams cover also situations where a different base theory (e.g. RCA_0 , which is RCA_0 with unrestricted induction) is used, or where only the first order consequences are considered.

Actually, the zoo is not peculiar to subsystems of second order arithmetic. For example, the study of weak forms of the Axiom of Choice and the relationships between them has a long tradition in set theory: [HR98] consists of a catalog of 383 forms of the Axiom of Choice and of their equivalent statements. Connected to the book, there is also the web page [How], which claims also to be able to produce zoo-like tables; unfortunately the site appears to be no longer maintained and, as of December 2015, the links are broken.

Mummert, Saadaoui and Sovine in [MSS15] introduced a framework for discussing the logic that is behind the web of implications and nonimplications in the reverse mathematics zoo. They called their system *s*-logic, introducing its syntax and semantics and proposing a tableaux system for satisfiability of sets of *s*-formulas, and inference systems for two fragments of *s*-logic (called \mathcal{F}_1 and \mathcal{F}_2 , with the first a subset of the second) that are important in the applications.

The present paper can be viewed as a continuation of [MSS15]. Our goal is to improve the systems introduced by Mummert, Saadaoui and Sovine and show how widespread automated theorem proving tools can be used to deal efficiently with *s*-logic. As a byproduct, our analysis also points out that, notwithstanding the fact that the semantics for *s*-logic borrows some ideas from the one for modal logic, *s*-logic is actually much closer to propositional logic than to modal logic.

Here is the plan of the paper. After reviewing *s*-logic, in Section 2 we make some observations about its semantics. Using these, in Section 3 we are able to simplify the tableaux system of Mummert, Saadaoui and Sovine. Our formulation brings it closer to the familiar tableaux systems for propositional logic, and thus, using an efficient implementation of the latter, leads to more efficient algorithms. Moreover, in Section 4, we improve also the treatment of the fragments \mathcal{F}_1 and \mathcal{F}_2 by proposing natural deduction systems for them. We also consider a new natural fragment of *s*-logic \mathcal{F}_3 , which includes \mathcal{F}_2 and for which we provide a sound and complete natural deduction system. In Section 5 we show how logical consequence between formulas of \mathcal{F}_2 (and hence of \mathcal{F}_1) can be treated by using standard propositional Prolog: this provides an efficient way of answering queries about whether a certain implication or nonimplication follows from a database of known zoo facts.

2. BASIC OBSERVATIONS ABOUT S-LOGIC

For the reader’s convenience, we start with a brief review of *s*-logic as introduced in [MSS15].

We start from a set of propositional variables and we build propositional formulas in the usual way, using the connectives \neg , \wedge , \vee , and \rightarrow . An *s*-formula is a formula of the form $A \rightarrow B$ or $A \not\rightarrow B$, where A and B are propositional formulas. The first type of *s*-formula is called positive or \rightarrow *s*-formula, the second one is negative or $\not\rightarrow$ *s*-formula. Notice that the

definition of s-formula is not recursive, and thus if α and β are s-formulas neither $\alpha \wedge \beta$ nor $\alpha \rightarrow \beta$ are s-formulas.

The intended meaning of $A \rightarrow B$ is that statement A implies statement B , over the fixed weak base theory. On the other hand $A \nrightarrow B$ asserts that $A \rightarrow B$ does not hold. In practice, this happens when we have a model of the base theory in which A holds and B does not (a counterexample to $A \rightarrow B$).

The semantics of s-logic is based on the notion of *frame*, which is just a nonempty set of valuations. Here by valuation we mean the usual notion for propositional logic, i.e. a function assigning to every propositional variable one of the truth values T and F .

A frame W *satisfies* the positive s-formula $A \rightarrow B$ if for every valuation $v \in W$ such that $v(A) = T$ we have also $v(B) = T$. W satisfies the negative s-formula $A \nrightarrow B$ if there exists a valuation $v \in W$ such that $v(A) = T$ and $v(B) = F$.

Once we have the notion of satisfaction we can introduce in the usual way notions such as *satisfiability* of a set of s-formulas Γ (there exists a frame satisfying every member of Γ) and *logical consequence* between a set of s-formulas Γ and a given s-formula α (every frame satisfying Γ satisfies also α): for the latter we use the notation $\Gamma \models_s \alpha$.

We point out that although \rightarrow and \nrightarrow (and their negations) are superficially similar, there are important difference between them. For example, if X and Y are propositional variables, the set of s-formulas $\{X \nrightarrow Y, Y \nrightarrow X\}$ is satisfiable (by a frame with two valuations), while the “corresponding” set of propositional formulas $\{\neg(X \rightarrow Y), \neg(Y \rightarrow X)\}$ is unsatisfiable. Expressing the same example in terms of logical consequence, we have that although $\neg(X \rightarrow Y) \models Y \rightarrow X$ in propositional logic, it is certainly not the case that $X \nrightarrow Y \models_s Y \rightarrow X$. Notice that in these examples we are using s-formulas from \mathcal{F}_1 .

Mummert, Saadaoui and Sovine introduced also the following fragments of s-logic:

Definition 1. *The fragment \mathcal{F}_1 , \mathcal{F}_2 of s-logic are:*

- \mathcal{F}_1 is the set of all s-formulas of the forms $X \rightarrow Y$ and $X \nrightarrow Y$, where X, Y are propositional variables;
- \mathcal{F}_2 is the set of all s-formulas of the forms $A \rightarrow Y$ and $A \nrightarrow Y$, where A is a nonempty conjunction of propositional variables and Y is a single propositional variable.

As pointed out in [MSS15], \mathcal{F}_1 captures the basic implications and nonimplications in reverse mathematics, while in \mathcal{F}_2 we can express also results such as the equivalence between Ramsey Theorem for pairs with two colors and the conjunction between the same theorem restricted to stable colorings and the cohesiveness principle. Notice that we do not need to consider also s-formulas with conjunctions of propositional variables after \rightarrow , as $\Gamma \models_s A \rightarrow X \wedge Y$ if and only if $\Gamma \models_s A \rightarrow X$ and $\Gamma \models_s A \rightarrow Y$, while $\Gamma \models_s A \nrightarrow X \wedge Y$ if and only if $\Gamma \models_s A \nrightarrow X$ or $\Gamma \models_s A \nrightarrow Y$.

We introduce another fragment of s-logic, which is a natural generalization of the fragment \mathcal{F}_2 , and captures some implications between members of the reverse mathematics zoo escaping \mathcal{F}_2 . Recall, for example, that the statement about the existence of iterates of continuous mappings of the closed unit interval into itself was proved in [FSY93] to be equivalent to the disjunction of weak König’s lemma and Σ_2^0 -induction.

Definition 2. \mathcal{F}_3 is the set of all s-formulas of the forms $C \rightarrow D$ and $C \nrightarrow D$, where C and D are a nonempty conjunction of propositional variables and a nonempty disjunction of propositional variables, respectively.

Here we do not need to consider also s-formulas with disjunctions of propositional variables before \neg , as $\Gamma \models_s X \vee Y \neg A$ if and only if $\Gamma \models_s X \neg A$ and $\Gamma \models_s Y \neg A$, while $\Gamma \models_s X \vee Y \not\models A$ if and only if $\Gamma \models_s X \not\models A$ or $\Gamma \models_s Y \not\models A$.

We now make a couple of useful basic observations about the semantics of s-logic which use the following definition.

Definition 3. Given a set of s-formulas Γ , the set of s-formulas Γ^+, Γ^- are defined as

$$\Gamma^+ := \{A \neg B : A \neg B \in \Gamma\}, \quad \Gamma^- := \{A \not\models B : A \not\models B \in \Gamma\},$$

while Γ_{prop}^+ is the set of propositional formulas

$$\Gamma_{prop}^+ := \{A \rightarrow B : A \neg B \in \Gamma\}.$$

Lemma 4. Let Γ be a set of s-formulas. The following are equivalent:

- (1) Γ is satisfiable;
- (2) the set of s-formulas

$$\Gamma^+ \cup \{A \not\models B\}$$

is satisfiable, for each $A \not\models B \in \Gamma^-$;

- (3) the set of propositional formulas

$$\Gamma_{prop}^+ \cup \{A, \neg B\}$$

is satisfiable (in the usual sense of propositional logic), for each $A \not\models B \in \Gamma^-$.

Proof. (1) implies (2) is immediate.

To prove that (2) implies (3) fix $A \not\models B \in \Gamma^-$. Since $\Gamma^+ \cup \{A \not\models B\}$ is satisfiable, there exists a frame W which validates this set of s-formulas; hence there exists a valuation $v \in W$ with $v(A) = T$, $v(B) = F$. Since $W \models X \neg Y$ for all $X \neg Y \in \Gamma^+$ we have that $v(X) = T$ implies $v(Y) = T$ for each such s-formula. Hence v satisfies the set of propositional formulas $\Gamma_{prop}^+ \cup \{A, \neg B\}$.

For (3) implies (1), suppose (3) holds, and for each $A \not\models B \in \Gamma^-$ let $w_{A \not\models B}$ be a valuation satisfying the set of propositional formulas $\Gamma_{prop}^+ \cup \{A, \neg B\}$. Let W be the frame consisting of all these valuations: $W = \{w_{A \not\models B} : A \not\models B \in \Gamma^-\}$. It is easily seen that W satisfies Γ . \square

Corollary 5. A set of s-formulas Γ is unsatisfiable if and only if there exists $A \not\models B \in \Gamma^-$ such that $\Gamma^+ \models A \neg B$. In particular, every set of positive s-formulas is satisfiable.

Lemma 4 suggests a fairly simple algorithm for the satisfiability problem for sets of s-formulas. In fact given the set of s-formulas Γ one needs only to check whether for each $A \not\models B \in \Gamma^-$ the set of propositional formulas $\Gamma_{prop}^+ \cup \{A, \neg B\}$ is satisfiable. Given the constant improvement in the efficiency of SAT-solvers (see e.g. [MSL14, ST13]), this is in fact a quite efficient way of dealing with the problem.

Corollary 6. The problem of satisfiability for a (finite) set of s-formulas has the same complexity of propositional satisfiability, i.e. it is NP-complete.

Proof. The problem is in NP because, if we fix a finite set of s-formulas Γ and set $n = |\Gamma|$ and $k = |\Gamma^+|$, using the last point of the previous Lemma, we can reduce the satisfiability of Γ to the satisfiability of $n - k$ sets of propositional formulas each of cardinality $k + 1$.

The problem is NP-complete because it essentially contains propositional satisfiability. \square

The previous corollary implies that with respect to complexity s-logic is more similar to propositional logic than to modal logic (recall that satisfiability for propositional logic is NP-complete, while satisfiability for the modal logic K is PSPACE-complete).

Next, we consider logical consequence among s-formulas.

Lemma 7. *Let Γ be a satisfiable set of s-formulas. For any propositional formulas A and B we have:*

- (i) $\Gamma \models_s A \rightarrow B$ if and only if $\Gamma^+ \models_s A \rightarrow B$ if and only if $\Gamma_{prop}^+ \models A \rightarrow B$;
- (ii) $\Gamma \models_s A \not\models B$ if and only if there exists an s-formula $E \not\models F \in \Gamma^-$ such that

$$\Gamma^+, A \rightarrow B \models_s E \rightarrow F,$$

if and only if there exists an s-formula $E \not\models F \in \Gamma^-$ such that

$$\Gamma_{prop}^+, A \rightarrow B \models E \rightarrow F.$$

Proof. (i) If $\Gamma \models_s A \rightarrow B$ then $\Gamma \cup \{A \not\models B\}$ is unsatisfiable. By Lemma 4, there exists $E \not\models F \in \Gamma^- \cup \{A \not\models B\}$ such that $\Gamma^+ \cup \{E \not\models F\}$ is unsatisfiable. Since Γ is satisfiable, $E \not\models F$ must be $A \not\models B$, and hence $\Gamma^+ \models_s A \rightarrow B$. The viceversa is obvious. The equivalence between $\Gamma^+ \models_s A \rightarrow B$ and $\Gamma_{prop}^+ \models A \rightarrow B$ follows easily from Lemma 4.

As for (ii), $\Gamma \models_s A \not\models B$ iff the set of s-formulas $\Gamma \cup \{A \rightarrow B\}$ is unsatisfiable iff (by Lemma 4) there exists $E \not\models F \in \Gamma^-$ such that $\Gamma^+ \cup \{A \rightarrow B\} \cup \{E \not\models F\}$ is unsatisfiable iff there exists $E \not\models F \in \Gamma^-$ such that $\Gamma^+, A \rightarrow B \models_s E \rightarrow F$ iff $\Gamma_{prop}^+, A \rightarrow B \models E \rightarrow F$. \square

The previous Lemma says that only positive s-formulas are needed to check whether a positive s-formula is logical consequence of a satisfiable set of s-formulas. Moreover, if only positive s-formulas are considered, their logic does not differ substantially from propositional logic, because \rightarrow behaves exactly as \rightarrow .

If we want to prove that a negative s-formula is logical consequence of a satisfiable set of s-formulas then differences with propositional logic do appear. The previous Lemma tells us that the collection of $\not\models$ s-formulas which are logical consequences of some $\not\models$ s-formulas (i.e. typically from the existence of different models showing that the implications fail) and some \rightarrow s-formulas is just the union of the consequences of a single $\not\models$ s-formula and the given set of \rightarrow s-formulas. In other words, having two models available simultaneously gives no new information. This might again suggest that s-logic is not substantially different from propositional logic. Nevertheless, the deductive meta-properties of s-logic and propositional logic differ, as showed by the following example.

Example 8. In propositional logic, if A, B, C, D are propositional variables and α is a formula, we have:

$$\Gamma, A \rightarrow C \models \alpha \quad \text{and} \quad \Gamma, B \rightarrow C \models \alpha \quad \text{then} \quad \Gamma, A \wedge B \rightarrow C \models \alpha.$$

This is not the case in s-logic because, for example:

$$A \not\models D, B \not\models D, A \rightarrow C \models_s C \not\models D,$$

$$A \not\models D, B \not\models D, B \rightarrow C \models_s C \not\models D$$

but

$$A \not\models D, B \not\models D, A \wedge B \rightarrow C \not\models_s C \not\models D.$$

In fact the set of s-formulas $\{A \not\models D, B \not\models D, A \wedge B \rightarrow C, C \rightarrow D\}$ is satisfied e.g. by the frame $W = \{v_1, v_2\}$ with $v_1(A) = v_2(B) = T$, $v_1(B) = v_2(A) = v_1(D) = v_2(D) = v_1(C) = v_2(C) = F$.

3. TABLEAUX FOR S-LOGIC

Another application of Lemma 4 regards the existence of a tableaux system to check unsatisfiability of finite set of s-formulas. In [MSS15], the authors introduce a tableaux system which keeps track of valuations in the syntax. For this reason the tableaux are unusual compared e.g. to the standard tableaux described in a textbook such as [BA12] (see §2.6, where they are called semantic tableaux). In fact to deal with strict non-implication the system considers not only s-formulas, but also so-called *world formulas*, that is, pairs (A, v) where A is a propositional formula and v represents a variable for a propositional evaluation. The tableaux system of [MSS15] contains e.g. the following rule (where Γ is a set of s- and world formulas, and v is new for Γ):¹

$$\frac{\Gamma, A \not\vdash B}{\Gamma, (A, v), (\neg B, v)}$$

The tableaux system of [MSS15] has also the peculiarity of not discharging the formulas which are used in a step (this is instead a common feature of tableaux systems for propositional logic, see [BA12, Algorithm 2.64]). This is motivated by the fact that positive s-formulas are in fact universal assertions about the collection of all possible worlds, and thus might be used again on a different world. However Lemma 4 shows that this precaution is superfluous, because the unsatisfiability of a set of s-formulas depends only on a single world, the one witnessing the satisfiability of one of the negative s-formulas that imply the unsatisfiability of the whole set.

A straightforward application of Lemma 4 leads to a more traditional tableaux system, which has the advantage of dealing only with propositional formulas, except for the first (root) step. This system can be described as follows. The rules of the system are given by the standard rules of a traditional tableaux system for propositional logic plus the $\not\vdash$ -rule, which is:

$$\frac{\Gamma, A \not\vdash B}{\Gamma_{prop}^+, A, \neg B},$$

subsuming the rule

$$\frac{\Gamma}{\Gamma_{prop}^+}$$

when $\Gamma^- = \emptyset$.

Notice that, starting from $\Gamma, A \not\vdash B, C \not\vdash D$, the $\not\vdash$ -rule allows to derive either $\Gamma_{prop}^+, A, \neg B$ or $\Gamma_{prop}^+, C, \neg D$.

Definition 9. A tableau for a set of s-formulas Γ is a finite tree T such that:

- (a) the root of T is labeled by Γ , while the inner nodes are labelled by sets of propositional formulas;
- (b) the label of the child of the root is obtained from the label of the root by an application of the $\not\vdash$ -rule;
- (c) the label of every other node is obtained from the label of its parent by one of the standard propositional tableaux inference rules (see e.g. [BA12, Algorithm 2.64]).

A path through a tableau is closed if it contains a node for which the label contains both A and $\neg A$ for some propositional formula A . A tableau is closed if every maximal branch is closed.

¹here and below we adopt the convention that the premisses of a rule are above their consequence, while in [MSS15] the reverse convention is adopted

Notice that, in contrast with the propositional case, a given set of s-formulas might have both closed and non-closed tableaux. In fact to obtain a closed tableau we must pick the “right” negative s-formula when we apply the \neg -rule to construct the child of the root, as is easily seen for the set of s-formulas $\{A \neg B, A \neg C\}$.

Applying Lemma 4 we immediately obtain:

Corollary 10. *A set of s-formula Γ is unsatisfiable if and only if there exists a tableau for Γ in which every branch is closed.*

The previous corollary is useful in practice, because to check satisfiability of s-formulas after the first step we use a standard tableaux system for propositional logic.

However, the tableaux system presented here and the one proposed in [MSS15] are hybrid systems, where s-formulas and propositional formulas coexist. Hence neither system is appropriate to study s-logic for itself, and compare its deductive properties with the ones of propositional logic, as we did in Example 8. What are the rules of s-logic, and can we have a calculus dealing exclusively with s-formulas? As in [MSS15], we answer these questions for some fragments of s-logic which are relevant to the practice of reverse mathematics. In our case these are the ones introduced in Definition 1 (considered also in [MSS15]) but also the fragment \mathcal{F}_3 introduced in Definition 2.

4. NATURAL DEDUCTIONS FOR FRAGMENTS OF S-LOGIC

Lemma 7 is especially useful when dealing with the fragments of Definitions 1 and 2. In [MSS15] sound and complete deductive systems for \mathcal{F}_1 and \mathcal{F}_2 are presented.

The system for \mathcal{F}_1 consists of the following axioms and rules:

- (Axiom): $X \neg X$, where X is a propositional variable;
- (HS): From $A \neg X$ and $X \neg Y$ deduce $A \neg Y$;
- (N): From $X \neg Y$, $X \neg W$ and $Z \neg Y$ deduce $W \neg Z$.

The system for \mathcal{F}_2 consists of the following axioms and rules:

- (Axiom): $X \neg X$, where X is a propositional variable;
- (W): From $A \neg Y$, deduce $B \neg Y$, where B is any conjunction such that every conjunct of A is also a conjunct of B ;
- (HS): From $X \wedge B \neg Y$ and $A \neg X$, deduce $A \wedge B \neg Y$;
- (N): From $A \neg X$, $A \wedge Z \neg X$, and $A \neg Y$ for each conjunct Y of B , deduce $B \neg Z$.

We propose natural deduction calculi for \mathcal{F}_2 and for \mathcal{F}_1 , differing from the systems in [MSS15] because of a simpler rule for negative s-formulas. We also introduce a natural deduction system for \mathcal{F}_3 . These systems are presented in a style similar to [HR04] (see §1.2.3 for a summary of natural deduction for propositional logic).

4.1. A Natural Deduction Calculus for \mathcal{F}_2 . The Natural Deduction Calculus for \mathcal{F}_2 has the following axioms and rules, where X, Y, Z, X_i, \dots are propositional variables, A, B, C, \dots are arbitrary (possibly empty) conjunctions of propositional variables, α is an arbitrary \mathcal{F}_2 formula, and Γ and Γ' are sets of \mathcal{F}_2 s-formulas:

(Axiom): $X \neg X$

$$(\text{conj1}): \frac{\frac{\Gamma}{\nabla} \quad A \neg Y}{A \wedge B \neg Y}$$

$$\text{(conj2): } \frac{\frac{\Gamma}{\nabla} \quad X_1 \wedge \dots \wedge X_n \rightarrow Y}{X_{i_1} \wedge \dots \wedge X_{i_k} \rightarrow Y},$$

where $\{X_{i_1}, \dots, X_{i_k}\} = \{X_1, \dots, X_n\}$ as *sets* of propositional variables.

$$\text{(trans): } \frac{\frac{\Gamma}{\nabla} \quad A \rightarrow Y \quad \frac{\Gamma'}{\nabla} \quad Y \wedge B \rightarrow Z}{A \wedge B \rightarrow Z}$$

$$(\perp): \frac{\frac{\Gamma}{\nabla} \quad A \rightarrow B \quad \frac{\Gamma}{\nabla} \quad A \not\rightarrow B}{\alpha}$$

For negative s-formulas we want a rule allowing to construct of a proof of $A \not\rightarrow X$ from hypothesis $\Gamma, C \not\rightarrow Y$, whenever we have a proof of $C \rightarrow Y$ from hypothesis $\Gamma, A \rightarrow X$:

$$\text{(neg): } \frac{\frac{\Gamma'}{\nabla} \quad C \not\rightarrow Y \quad \frac{\Gamma, [A \rightarrow X]}{\nabla} \quad C \rightarrow Y}{A \not\rightarrow X}$$

Let $\Gamma \triangleright_{\mathcal{F}_2} \alpha$ denote the existence of a natural deduction proof (in the system just described) of the \mathcal{F}_2 s-formula α from hypothesis in the set of \mathcal{F}_2 s-formulas Γ .

Example 11. Here is a deduction showing that

$$A \not\rightarrow X, A \wedge Z \rightarrow X, A \rightarrow Y_1, \dots, A \rightarrow Y_n \triangleright_{\mathcal{F}_2} Y_1 \wedge \dots \wedge Y_n \not\rightarrow Z,$$

corresponding to rule (N) in the \mathcal{F}_2 system of [MSS15]:

$$\frac{\frac{\frac{A \rightarrow Y_1 \quad [Y_1 \wedge \dots \wedge Y_n \rightarrow Z]}{A \wedge Y_2 \wedge \dots \wedge Y_n \rightarrow Z}}{A \wedge Y_3 \wedge \dots \wedge Y_n \rightarrow Z}}{\vdots} \quad \frac{A \rightarrow Y_n \quad A \wedge Y_n \rightarrow Z}{A \rightarrow Z} \quad \frac{A \wedge Z \rightarrow X}{A \rightarrow X} \quad \frac{A \not\rightarrow X \quad Y_1 \wedge \dots \wedge Y_n \not\rightarrow Z}{A \not\rightarrow X}$$

Here double lines indicate combined applications of (conj2) and (trans), the top step consists of an application of (trans), and the last step is an application of (neg).

One can easily prove that all \mathcal{F}_2 rules are sound with respect to s-logical consequence. As for completeness, we divide the proof into cases, depending on the satisfiability of the set of premisses Γ .

Lemma 12. *If Γ is a satisfiable set of \mathcal{F}_2 s-formulas and α is a \mathcal{F}_2 s-formula such that $\Gamma \models_s \alpha$ then $\Gamma \triangleright_{\mathcal{F}_2} \alpha$.*

Proof. To prove the Lemma we rely on Theorem 17 from [MSS15], which says that if Γ is a satisfiable² and $\Gamma \models_s \alpha$ then α is derivable from Γ using the rules (Axiom), (W), (HS), and (N). Hence, to show that α is derivable in our system it is enough to show the existence of natural deduction proofs for rules (W), (HS), and (N). The only nontrivial case is rule (N), which is dealt with in Example 11. \square

To finish the completeness proof for $\triangleright_{\mathcal{F}_2}$, we have to consider the case when Γ is unsatisfiable, where we need to prove that $\Gamma \triangleright_{\mathcal{F}_2} \alpha$, for any \mathcal{F}_2 s-formula α .

Lemma 13. *If Γ is unsatisfiable, then for any \mathcal{F}_2 s-formula α we have $\Gamma \triangleright_{\mathcal{F}_2} \alpha$.*

Proof. By Corollary 5, if Γ is unsatisfiable then there exists $A \not\vdash B \in \Gamma^-$ such that $\Gamma^+ \models_s A \rightarrow B$. Since Γ^+ is satisfiable (again by Corollary 5), by Lemma 12 we have $\Gamma^+ \triangleright_{\mathcal{F}_2} A \rightarrow B$. Hence $\Gamma \triangleright_{\mathcal{F}_2} A \rightarrow B$, and $\Gamma \triangleright_{\mathcal{F}_2} \alpha$ follows by rule (\perp). \square

Putting all the results of this subsection together, we obtain:

Theorem 14. *If Γ is a set of \mathcal{F}_2 s-formulas and α is a \mathcal{F}_2 s-formula, then*

$$\Gamma \models_s \alpha \quad \Leftrightarrow \quad \Gamma \triangleright_{\mathcal{F}_2} \alpha.$$

4.2. A Natural Deduction Calculus for \mathcal{F}_1 . The Natural Deduction Calculus for \mathcal{F}_1 has the following axioms and rules (where X, Y, Z are propositional variables, α is a \mathcal{F}_1 s-formula, and Γ and Γ' are sets of \mathcal{F}_1 s-formulas):

(Axiom): $X \rightarrow X$

$$\text{(trans):} \quad \frac{\begin{array}{c} \Gamma \\ \nabla \\ X \rightarrow Y \end{array} \quad \begin{array}{c} \Gamma' \\ \nabla \\ Y \rightarrow Z \end{array}}{X \rightarrow Z}$$

$$\text{(neg):} \quad \frac{\begin{array}{c} \Gamma' \\ \nabla \\ Y \not\vdash Z \end{array} \quad \begin{array}{c} \Gamma, [X \rightarrow Y] \\ \nabla \\ Y \rightarrow Z \end{array}}{X \not\vdash Y}$$

$$(\perp): \quad \frac{\begin{array}{c} \Gamma \\ \nabla \\ A \rightarrow B \end{array} \quad \begin{array}{c} \Gamma' \\ \nabla \\ A \not\vdash B \end{array}}{\alpha}$$

Let $\Gamma \triangleright_{\mathcal{F}_1} \alpha$ denotes the existence of a natural deduction proof (in the system just described) of the \mathcal{F}_1 s-formula α from hypothesis in the set of \mathcal{F}_1 s-formulas Γ .

Example 15. Here is a deduction showing that

$$X \not\vdash Y, X \rightarrow W, Z \rightarrow Y \triangleright_{\mathcal{F}_1} W \not\vdash Z,$$

corresponding to rule (N) in the \mathcal{F}_1 system of [MSS15]:

²actually, the hypothesis in [MSS15] is that Γ is consistent, but an inspection of the proof reveals that the right hypothesis is the one of satisfiability.

$$\frac{\frac{X \rightarrow W \quad [W \rightarrow Z]}{X \rightarrow Z} \quad \frac{Z \rightarrow Y}{X \rightarrow Y} \quad X \not\rightarrow Y}{W \not\rightarrow Z}$$

Here we employed (trans) twice and (neg) for the last step.

As for the case of the \mathcal{F}_2 system, the soundness of $\triangleright_{\mathcal{F}_1}$ is easily proved, and left to the reader. For completeness, we may follow the same line of the completeness proof for $\triangleright_{\mathcal{F}_2}$, dividing the proof into cases, depending on whether Γ is a satisfiable set of \mathcal{F}_1 s-formulas or not. The case where Γ is satisfiable can be dealt using Theorem 20 from [MSS15], and consists in proving the \mathcal{F}_1 rules of [MSS15] in our system. The only nontrivial case is rule (N), which is dealt with in Example 15. In the case where Γ is unsatisfiable, we may proceed using rule \perp as we did for $\triangleright_{\mathcal{F}_2}$. Hence:

Theorem 16. *If Γ is a set of \mathcal{F}_1 s-formulas and α is a \mathcal{F}_1 s-formula, then*

$$\Gamma \models_s \alpha \quad \Leftrightarrow \quad \Gamma \triangleright_{\mathcal{F}_1} \alpha.$$

4.3. A Natural Deduction Calculus for \mathcal{F}_3 . We now consider the fragment \mathcal{F}_3 introduced in Definition 2. In considering an \mathcal{F}_3 s-formula $C \rightarrow D$ or $C \not\rightarrow D$ we denote by C_i a propositional variable which is a C -conjunct and by D_j a propositional variable which is a D -disjunct.

In order to capture derivability in fragment \mathcal{F}_3 , we extend our natural deduction calculus for \mathcal{F}_2 with the following two rules:

$$\text{(disj1):} \quad \frac{\Gamma \quad \nabla \quad \frac{A \rightarrow B}{A \rightarrow D}}{A \rightarrow D}$$

where $\{B_1, \dots, B_n\} \subseteq \{D_1, \dots, D_h\}$ as *sets* of propositional variables.

$$\text{(disj2):} \quad \frac{\frac{\Gamma \quad \nabla \quad A \rightarrow B}{A \rightarrow B} \quad \frac{\Gamma, [A \rightarrow B_1] \quad \nabla \quad C \rightarrow E}{C \rightarrow E} \quad \dots \quad \frac{\Gamma, [A \rightarrow B_n] \quad \nabla \quad C \rightarrow E}{C \rightarrow E}}{C \rightarrow E}$$

where $B = B_1 \vee \dots \vee B_n$ and Γ is a set of positive s-formulas.

Lemma 17. *Rules (disj1) and (disj2) are sound in s-logic.*

Proof. Soundness of rule (disj1) is immediate.

As for rule (disj2), suppose Γ is a positive set of \mathcal{F}_3 s-formulas and $B = B_1 \vee \dots \vee B_n$ is such that:

- $\Gamma \models_s A \rightarrow B$;
- $\Gamma, A \rightarrow B_i \models_s C \rightarrow E$ for each $i = 1, \dots, n$.

We want to prove that $\Gamma \models_s C \rightarrow E$. Since Γ contains only positive s-formulas, by Corollary 5 each set $\Gamma, A \rightarrow B_i$ is satisfiable. Hence we may apply Lemma 7 obtaining:

$$\Gamma_{prop}^+, A \rightarrow B_i \models C \rightarrow E.$$

Similarly we obtain $\Gamma_{prop}^+ \models A \rightarrow B$, that is $\Gamma_{prop}^+, A \models B$. By propositional reasoning it follows that $\Gamma_{prop}^+ \models C \rightarrow E$. Hence, by Lemma 7 again, $\Gamma \models_s C \rightarrow E$. \square

Notice that the restriction to positive set of s-formulas Γ in rule (disj2) is necessary because without this hypothesis the rule is no longer sound. To see this consider e.g. the set

$$\Gamma = \{A \rightarrow B_1 \vee B_2, A \not\rightarrow B_1, A \not\rightarrow B_2\}.$$

Γ is satisfiable, while each set $\Gamma \cup \{A \rightarrow B_i\}$, for $i = 1, 2$, is unsatisfiable. It follows that any formula $C \rightarrow D$ (with C, D new for Γ) is a s-consequence of both sets $\Gamma \cup \{A \rightarrow B_i\}$. Moreover, $\Gamma \models_s A \rightarrow B_1 \vee B_2$, but $C \rightarrow D$ is not a s-consequence of Γ .

We denote \mathcal{F}_3 -derivability by $\triangleright_{\mathcal{F}_3}$. In proving the completeness of the \mathcal{F}_3 system we shall use also the following three rules, that will be shown to be derivable in our system in the next Lemma.

$$(r_2): \frac{\begin{array}{c} \Gamma \\ \nabla \\ B \rightarrow A \end{array} \quad \begin{array}{c} \Gamma \\ \nabla \\ C \rightarrow B_1 \end{array} \quad \cdots \quad \begin{array}{c} \Gamma \\ \nabla \\ C \rightarrow B_n \end{array}}{C \rightarrow A}$$

where $B = B_1 \wedge \cdots \wedge B_n$.

$$(r_3): \frac{\begin{array}{c} \Gamma \\ \nabla \\ D \rightarrow E \end{array} \quad \begin{array}{c} \Gamma \\ \nabla \\ D \wedge E_1 \rightarrow F \end{array} \quad \cdots \quad \begin{array}{c} \Gamma \\ \nabla \\ D \wedge E_n \rightarrow F \end{array}}{D \rightarrow F}$$

where $E = E_1 \vee \cdots \vee E_n$.

$$(disj2gen): \frac{\begin{array}{c} \Gamma \\ \nabla \\ A^1 \rightarrow B^1 \end{array} \quad \cdots \quad \begin{array}{c} \Gamma \\ \nabla \\ A^n \rightarrow B^n \end{array} \quad \cdots \quad \begin{array}{c} \Gamma, [A^1 \rightarrow B_{h_1}^1, \dots, A^n \rightarrow B_{h_n}^n] \\ \nabla \\ C \rightarrow E \end{array}}{C \rightarrow E}$$

In (disj2gen), we require Γ to be a set of positive s-formulas, and we have a premise

$$\begin{array}{c} \Gamma, A^1 \rightarrow B_{h_1}^1, \dots, A^n \rightarrow B_{h_n}^n \\ \nabla \\ C \rightarrow E \end{array}$$

for every choice of indices h_1, \dots, h_n such that $B_{h_i}^i$ is a disjunct of B^i .

Lemma 18. (r_2) is a derived rule in the \mathcal{F}_2 system, while (r_3) and $(disj2gen)$ are derived rules in the \mathcal{F}_3 system.

Proof. First, we provide a proof for (r_2) in the \mathcal{F}_2 system.

$$\frac{\begin{array}{c} \Gamma \\ \nabla \\ B_1 \wedge \dots \wedge B_n \rightarrow A \end{array} \quad \begin{array}{c} \Gamma \\ \nabla \\ C \rightarrow B_1 \end{array} \quad \begin{array}{c} \Gamma \\ \nabla \\ C \rightarrow B_2 \end{array}}{\frac{C \wedge B_2 \wedge \dots \wedge B_n \rightarrow A \quad C \rightarrow B_2}{C \wedge B_3 \wedge \dots \wedge B_n \rightarrow A} \quad \begin{array}{c} \Gamma \\ \nabla \\ C \rightarrow B_n \end{array}}{\frac{C \wedge B_n \rightarrow A \quad C \rightarrow B_n}{C \rightarrow A}}$$

where in the first step we apply (trans) and then, in correspondence of each double line, we use a combination of applications of (trans) and (conj2).

We now show how to derive (r_3) in the \mathcal{F}_3 system.

$$\frac{\frac{\Gamma}{\nabla} \quad \frac{[D \rightarrow E_1] \quad \frac{\Gamma}{\nabla} \quad D \wedge E_1 \rightarrow F}{D \rightarrow F} \quad \cdots \quad \frac{[D \rightarrow E_n] \quad \frac{\Gamma}{\nabla} \quad D \wedge E_n \rightarrow F}{D \rightarrow F}}{D \rightarrow F}$$

Again, double lines indicate a combination of applications of (trans) and (conj2), while in the final step we use (disj2).

As for rule (disj2gen), suppose $B^1 = B_1^1 \vee \cdots \vee B_h^1$. We can apply rule (disj2) to

$$\Gamma \triangleright_{\mathcal{F}_3} A^1 \rightarrow B^1$$

and all premisses of the form

$$\Gamma, A^1 \rightarrow B_j^1, A^2 \rightarrow B_{h_2}^1, \dots, A^n \rightarrow B_{h_n}^1 \triangleright_{\mathcal{F}_3} C \rightarrow E,$$

for $j = 1, \dots, h$, obtaining, for all choices of indices h_2, \dots, h_n such that $B_{h_i}^i$ is a disjunct of B^i , that

$$\Gamma, A^2 \rightarrow B_{h_2}^2, \dots, A^n \rightarrow B_{h_n}^n \triangleright_{\mathcal{F}_3} C \rightarrow E;$$

In other words, we succeeded in eliminating $A^1 \rightarrow B^1$ from the premisses. In the same way, by applying (disj2) we can successively eliminate $A^2 \rightarrow B^2, \dots, A^n \rightarrow B^n$, eventually deriving $\Gamma \triangleright_{\mathcal{F}_3} C \rightarrow E$, as desired. \square

In order to prove the completeness of \mathcal{F}_3 -derivability we need a preliminary Lemma.

Lemma 19. *Suppose Γ is a set of positive \mathcal{F}_3 s-formulas such that $\Gamma \not\vdash_{\mathcal{F}_3} C \rightarrow E$. Then there exists a set of positive \mathcal{F}_3 s-formulas Δ , closed under $\triangleright_{\mathcal{F}_3}$, such that:*

- $\Delta \supseteq \Gamma$;
- $\Delta \not\vdash_{\mathcal{F}_3} C \rightarrow E$;
- for all positive \mathcal{F}_3 s-formulas $A \rightarrow B \in \Delta$ there exists i such that $A \rightarrow B_i \in \Delta$.

Proof. Without loss of generality, we may suppose that Γ is closed under $\triangleright_{\mathcal{F}_3}$. Let $\{\alpha_1, \alpha_2, \dots\}$ be an enumeration of the positive \mathcal{F}_3 s-formulas, with $\alpha_j = A^j \rightarrow B^j$.

We claim that there exists a sequence $\Gamma_0 = \Gamma, \Gamma_1, \dots, \Gamma_n, \dots$ of sets of positive \mathcal{F}_3 s-formulas, each closed under $\triangleright_{\mathcal{F}_3}$, with the following properties:

- $\Gamma_n \not\vdash_{\mathcal{F}_3} C \rightarrow E$;
- if, for $j \leq n$, $\Gamma_n \triangleright_{\mathcal{F}_3} \alpha_j$, then there exists h such that $A^j \rightarrow B_h^j \in \Gamma_{n+1}$.

We start by setting $\Gamma_0 = \Gamma$. Suppose now we already defined Γ_n such that $\Gamma_n \not\vdash_{\mathcal{F}_3} C \rightarrow E$. Let $j_1, \dots, j_h \leq n$ be the list of all indices up to n such that $\Gamma_n \triangleright_{\mathcal{F}_3} \alpha_{j_i}$. Then there must exist a choice of indices h_{j_1}, \dots, h_{j_h} such that $B_{h_{j_i}}^{j_i}$ is a disjunct of B^{j_i} , and

$$\Gamma_n, A^{j_1} \rightarrow B_{h_{j_1}}^{j_1}, \dots, A^{j_h} \rightarrow B_{h_{j_h}}^{j_h} \not\vdash_{\mathcal{F}_3} C \rightarrow E.$$

In fact, if this were not the case, using rule (disj2gen), we would obtain that $\Gamma_n \triangleright_{\mathcal{F}_3} C \rightarrow E$. We fix such h_{j_1}, \dots, h_{j_h} and let Γ_{n+1} be the closure of $\Gamma_n \cup \{A^1 \rightarrow B_{h_{j_1}}^1, \dots, A^n \rightarrow B_{h_{j_h}}^n\}$ under $\triangleright_{\mathcal{F}_3}$. This proves the claim.

Finally, it is straightforward to check that $\Delta = \bigcup_n \Gamma_n$ has the required properties. \square

We split the proof of the completeness of $\triangleright_{\mathcal{F}_3}$ into cases, depending on the satisfiability of Γ and on the type of the formula to be derived. We start with:

Lemma 20. *Suppose Γ is a satisfiable set of \mathcal{F}_3 s-formulas and $C \rightarrow E$ is a positive \mathcal{F}_3 s-formula such that $\Gamma \models_s C \rightarrow E$. Then $\Gamma \triangleright_{\mathcal{F}_3} C \rightarrow E$.*

Proof. We reason by contradiction. If $\Gamma \not\triangleright_{\mathcal{F}_3} C \rightarrow E$ then $\Gamma^+ \not\triangleright_{\mathcal{F}_3} C \rightarrow E$, either. By applying the previous Lemma to Γ^+ we find a set of positive \mathcal{F}_3 s-formulas $\Delta \supseteq \Gamma^+$, closed under $\triangleright_{\mathcal{F}_3}$, such that

$$\Delta \not\triangleright_{\mathcal{F}_3} C \rightarrow E,$$

and for all \mathcal{F}_3 s-formulas $A \rightarrow B$, if $A \rightarrow B \in \Delta$ then there exists i with $A \rightarrow B_i \in \Delta$.

Let w be the valuation defined by setting, for each propositional variable X :

$$w(X) = \begin{cases} T & \text{if } C \rightarrow X \in \Delta; \\ F & \text{if } C \rightarrow X \notin \Delta. \end{cases}$$

We claim that $w(\Delta) = T$, and $w(C \rightarrow E) = F$.

If $B \rightarrow A \in \Delta$ and $w(B) = T$, then, since $B = B_1 \wedge \dots \wedge B_n$, we have $w(B_i) = T$ for all i . By definition of w , for all i it holds $C \rightarrow B_i \in \Delta$, and by rule (r_2) we obtain $C \rightarrow A \in \Delta$. By the property of Δ there exists i such that $C \rightarrow A_i \in \Delta$. Hence $w(A_i) = T$ and therefore $w(A) = T$ as well. This proves that $w(B \rightarrow A) = T$, for all $B \rightarrow A \in \Delta$.

Let us now show that $w(C \rightarrow E) = F$. Since $w(C) = T$, it suffices to prove that $w(E_i) = F$, for all i . If $w(E_i) = T$ for some i , then $C \rightarrow E_i \in \Delta$ and $C \rightarrow E \in \Delta$ would follow by rule (disj1) .

Having established the claim, we conclude the proof as follows. For all negative \mathcal{F}_3 s-formulas $\alpha = A \nrightarrow B \in \Gamma$, let v_α be a valuation such that $v_\alpha(\Gamma) = T$, $v_\alpha(A) = T$ and $v_\alpha(B) = F$. Such a v_α exists, because by hypothesis Γ is satisfiable. Then the frame $W = \{w\} \cup \{v_\alpha : \alpha \in \Gamma^-\}$ is such that $W \models \Gamma$ and $W \not\models C \rightarrow E$, contradicting our hypothesis. \square

Next, we consider the case in which Γ is satisfiable, but the formula to be derived is negative.

Lemma 21. *Suppose Γ is a satisfiable set of \mathcal{F}_3 s-formulas and $C \nrightarrow G$ is a negative \mathcal{F}_3 s-formula such that $\Gamma \models_s C \nrightarrow G$. Then $\Gamma \triangleright_{\mathcal{F}_3} C \nrightarrow G$.*

Proof. This proof follows the corresponding proof in [MSS15] with minor adjustments. We reason again by contradiction supposing (without loss of generality) that Γ closed under $\triangleright_{\mathcal{F}_3}$ and $C \nrightarrow G \notin \Gamma$. For any $\alpha = D \nrightarrow E \in \Gamma^-$, we will find a valuation w_α with $w_\alpha(\Gamma^+) = T$, $w_\alpha(D) = T$ and $w_\alpha(E) = F$, and either $w_\alpha(C) = F$ or $w_\alpha(G) = T$. Once this is done, we may set

$$W = \{w_\alpha : \alpha \in \Gamma^-\}$$

and find a contradiction, since W is a frame satisfying Γ but failing to satisfy $C \nrightarrow G$.

Fix $\alpha = D \nrightarrow E \in \Gamma^-$. Since Γ is satisfiable, there exists a valuation w with $w(\Gamma^+) = T$, $w(D) = T$, and $w(E) = F$. In order to find w_α we may suppose that all the valuations w with these properties satisfy also $w(C) = T$ (otherwise we may choose such a w for w_α). Consider the set of positive s-formulas $\Gamma^+ \cup \{C \rightarrow G\}$. Then

$$\Gamma^+ \cup \{C \rightarrow G\} \not\triangleright_{\mathcal{F}_3} D \rightarrow E,$$

otherwise, since $\Gamma \triangleright_{\mathcal{F}_3} D \nrightarrow E$, we would have $\Gamma \triangleright_{\mathcal{F}_3} C \nrightarrow G$ be the (neg) rule. By Lemma 19 there exists a set of positive formulas $\Delta \supseteq \Gamma^+ \cup \{C \rightarrow G\}$, closed under $\triangleright_{\mathcal{F}_3}$, such that

$\Delta \not\vdash_{\mathcal{F}_3} D \rightarrow E$, and for all A, B , if $A \rightarrow B \in \Delta$ then there exists i with $A \rightarrow B_i \in \Delta$. We claim that $D \rightarrow C_i \in \Delta$ for every i . To see this, we consider the valuation w defined as

$$w(X) = \begin{cases} T & \text{if } D \rightarrow X \in \Delta; \\ F & \text{if } D \rightarrow X \notin \Delta. \end{cases}$$

As in Lemma 20, it is not difficult to check that $w(\Delta) = T$, and $w(D \rightarrow E) = F$. By the previous hypothesis, we have $w(C) = T$, that is, $w(C_i) = T$ for all i . By definition of w this implies $D \rightarrow C_i \in \Delta$.

Next, consider the valuation v_i defined as

$$v_i(X) = \begin{cases} T & \text{if } D \wedge G_i \rightarrow X \in \Delta; \\ F & \text{if } D \wedge G_i \rightarrow X \notin \Delta. \end{cases}$$

We claim that there exists i with $v_i(E) = F$. Otherwise, we have $v_i(E) = T$, for all i . This means that for all i there exists j with $v_i(E_j) = T$, that is, by definition of v_i , $D \wedge G_i \rightarrow E_j \in \Delta$. It follows that, for all i , $D \wedge G_i \rightarrow E \in \Delta$. Consider the following natural deduction, which uses first (r_2) and then (r_3) ;

$$\frac{\begin{array}{ccccccc} \Delta & & \Delta & & \Delta & & \Delta \\ \nabla & & \nabla & & \nabla & & \nabla \\ C \rightarrow G & D \rightarrow C_1 & \cdots & D \rightarrow C_k & & & \end{array}}{D \rightarrow G} \quad \frac{\begin{array}{ccccccc} \Delta & & \Delta & & \Delta & & \Delta \\ \nabla & & \nabla & & \nabla & & \nabla \\ D \wedge G_1 \rightarrow E & \cdots & D \wedge G_n \rightarrow E & & & & \end{array}}{D \rightarrow E}$$

This contradicts $\Delta \not\vdash D \rightarrow E$.

Thus we can pick i such that $v_i(E) = F$. We have $v_i(D) = T$, $v_i(E) = F$, and $v_i(G) = T$, since $D \wedge G_i \rightarrow G_i \in \Delta$ and G is a disjunction. Moreover, as before, $v_i(\Delta) = T$: if $A \rightarrow B \in \Delta$ and $v_i(A) = T$, then $D \wedge G_i \rightarrow A_j \in \Delta$, for all j . By rule (r_2) we obtain $D \wedge G_i \rightarrow B \in \Delta$, and by the properties of Δ there exists h with $D \wedge G_i \rightarrow B_h \in \Delta$; hence, $v_i(B_h) = T$, and $v_i(B) = T$. It follows that $v_i(\Gamma^+) = T$, and we may choose such a v_i as w_α , finishing the proof. \square

The two previous results prove that, if Γ is a satisfiable set of \mathcal{F}_3 s-formulas, then for any \mathcal{F}_3 s-formula α such that $\Gamma \models_s \alpha$ we have $\Gamma \triangleright_{\mathcal{F}_3} \alpha$.

To finish the completeness proof for $\triangleright_{\mathcal{F}_3}$, we still have to consider the case when Γ is unsatisfiable. In this case we have to prove that $\Gamma \triangleright_{\mathcal{F}_3} \alpha$, for any \mathcal{F}_3 s-formula α , and we may repeat the proof of Lemma 13. Hence:

Lemma 22. *If Γ is unsatisfiable, then for any \mathcal{F}_3 s-formula α we have $\Gamma \triangleright_{\mathcal{F}_3} \alpha$.*

Putting all results of this section together, we obtain:

Theorem 23. *If Γ is a set of \mathcal{F}_3 s-formulas and α is a \mathcal{F}_3 s-formula, then*

$$\Gamma \models_s \alpha \quad \Leftrightarrow \quad \Gamma \triangleright_{\mathcal{F}_3} \alpha.$$

5. \mathcal{F}_2 AND PROLOG

In this section we show how standard Prolog may be used to deal with logical consequence in \mathcal{F}_2 . Since some readers might be unfamiliar with Prolog, we recall here the basic constructs of this programming language (restricting ourselves to the propositional setting), following [NS97] (see §I.10, and especially Definition 10.4).

Propositional Prolog deals with *Horn clauses* (finite sets of literals containing at most one positive literal), thought as disjunctions of their elements. When the Horn clause contains (exactly) one positive literal $\{Y, \neg X_1, \dots, \neg X_n\}$ it is a *program clause* and we write $Y :- X_1, \dots, X_n$. If $n > 0$ we think that the program clause is representing $X_1 \wedge \dots \wedge X_n \rightarrow Y$ and we call it a *rule*. If in the program clause we have $n = 0$ it is a *fact* and we write $Y :-$. If the Horn clause has only negative literals $\{\neg X_1, \dots, \neg X_n\}$ we call it a *goal* and write $:- X_1, \dots, X_n$. A *Prolog program* is a set of program clauses.

The typical situation is that we are given a Prolog program, and we want to know whether a conjunction of facts Y_1, \dots, Y_k is logical consequence of the given facts and rules. To this end we add the goal $\{\neg Y_1, \dots, \neg Y_k\}$ to the program and ask whether the resulting set of Horn clauses is unsatisfiable. This is the case if and only if applying the resolution rule repeatedly to the elements of the set starting with the goal we obtain the empty clause. Prolog works by searching all possible ways of applying the resolution rule with these constraints: if the search succeeds we have a *refutation* of the goal from the program.

We can now go back to our study of the \mathcal{F}_2 fragment of s-logic.

Definition 24. Given a set Γ of \mathcal{F}_2 s-formulas, define *Prolog*(Γ^+) to be the following Prolog program:

$$\text{Prolog}(\Gamma^+) = \{Z :- A_1, \dots, A_n \mid A_1 \wedge \dots \wedge A_n \rightarrow Z \in \Gamma^+\}.$$

We have:

Lemma 25. Let Γ be a set of \mathcal{F}_2 s-formulas and $A \rightarrow Y$ be a \mathcal{F}_2 s-formula, where $A = A_1 \wedge \dots \wedge A_n$.

(i) $\Gamma \models_s A \rightarrow Y$ if and only if there is a refutation of the goal $:- Y$ from the Prolog program

$$\text{Prolog}(\Gamma^+) \cup \{A_1 :-, \dots, A_n :-\};$$

(ii) $\Gamma \models_s A \not\models Y$ if and only if there exists $Z_1 \wedge \dots \wedge Z_n \not\models W \in \Gamma^-$ and a refutation of the goal $:- W$ from the Prolog program

$$\text{Prolog}(\Gamma^+) \cup \{Y :- A_1, \dots, A_n, Z_1 :-, \dots, Z_n :-\}.$$

Proof. (i) From Lemma 7.i we have that $\Gamma \models A \rightarrow Y$ if and only if $\Gamma_{prop}^+, A \models Y$. Since Γ is a set of \mathcal{F}_2 -formulas, the elements in Γ_{prop}^+ are (essentially) rules, while A is equivalent to the conjunction of the facts $A_1 :-, \dots, A_n :-$. Since Y is a positive literal, the equivalence follows from the completeness of Propositional Prolog.

(ii) From Lemma 7.ii we have that $\Gamma \models A \not\models Y$ if and only if there exists $Z_1 \wedge \dots \wedge Z_n \not\models W \in \Gamma^-$ such that

$$\Gamma_{prop}^+, A \rightarrow Y, Z_1 \wedge \dots \wedge Z_n \models W.$$

As before, the equivalence follows from interpreting this logical consequence in terms of Prolog and applying the completeness of Propositional Prolog. \square

Lemma 25 suggests an efficient way of checking logical consequence between \mathcal{F}_2 s-formulas based on a well-known programming language such as Prolog, and actually only for the special case of goals consisting of a single literal.

REFERENCES

- [BA12] Mordechai Ben-Ari. *Mathematical logic for computer science*. Springer, London, third edition, 2012.
- [CMS04] Peter Cholak, Alberto Marcone, and Reed Solomon. Reverse mathematics and the equivalence of definitions for well and better quasi-orders. *J. Symbolic Logic*, 69(3):683–712, 2004.
- [Dzh] Damir D. Dzhabarov. Reverse Mathematics Zoo. <http://rmzoo.math.uconn.edu/>.

- [FSY93] Harvey Friedman, Stephen G. Simpson, and Xiaokang Yu. Periodic points and subsystems of second-order arithmetic. *Ann. Pure Appl. Logic*, 62(1):51–64, 1993. Logic Colloquium '89 (Berlin).
- [Hir15] Denis R. Hirschfeldt. *Slicing the truth*, volume 28 of *Lecture Notes Series. Institute for Mathematical Sciences. National University of Singapore*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2015. On the computable and reverse mathematics of combinatorial principles, Edited and with a foreword by Chitat Chong, Qi Feng, Theodore A. Slaman, W. Hugh Woodin and Yue Yang.
- [How] Paul Howard. Consequences of the Axiom of Choice Project Homepage. <http://www.math.purdue.edu/~hrubin/JeanRubin/Papers/conseq.html>.
- [HR98] Paul Howard and Jean E. Rubin. *Consequences of the axiom of choice*, volume 59 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 1998. With 1 IBM-PC floppy disk (3.5 inch; WD).
- [HR04] Michael Huth and Mark Ryan. *Logic in computer science*. Cambridge University Press, Cambridge, second edition, 2004. Modelling and reasoning about systems.
- [HS07] Denis R. Hirschfeldt and Richard A. Shore. Combinatorial principles weaker than Ramsey's theorem for pairs. *J. Symbolic Logic*, 72(1):171–206, 2007.
- [Mar07] Alberto Marcone. Interval orders and reverse mathematics. *Notre Dame J. Formal Logic*, 48(3):425–448 (electronic), 2007.
- [Mon11] Antonio Montalbán. Open questions in reverse mathematics. *Bull. Symbolic Logic*, 17(3):431–454, 2011.
- [MSL14] Joao Marques-Silva and Ines Lynce. SAT solvers. In *Tractability*, pages 331–349. Cambridge Univ. Press, Cambridge, 2014.
- [MSS15] Carl Mummert, Alaeddine Saadaoui, and Sean Sovine. The modal logic of Reverse Mathematics. *Arch. Math. Logic*, 54(3-4):425–437, 2015.
- [NS97] Anil Nerode and Richard A. Shore. *Logic for applications*. Graduate Texts in Computer Science. Springer-Verlag, New York, second edition, 1997.
- [Pat] Ludovic Patey. Manually maintained Zoo. <http://www.ludovicpatey.com/media/research/zoo.pdf>.
- [Sim09] Stephen G. Simpson. *Subsystems of second order arithmetic*. Perspectives in Logic. Cambridge University Press, Cambridge; Association for Symbolic Logic, Poughkeepsie, NY, second edition, 2009.
- [ST13] Uwe Schöning and Jacobo Torán. *The satisfiability problem*, volume 3 of *Mathematik für Anwendungen [Mathematics for Applications]*. Lehmanns Media, Berlin, 2013. Algorithms and analyses, Translated from the 2011 German original.

DIPARTIMENTO DI MATEMATICA E INFORMATICA, UNIVERSITÀ DI UDINE, VIALE DELLE SCIENZE 206, 33100 UDINE, ITALY
 E-mail address: giovanna.dagostino@uniud.it

DIPARTIMENTO DI MATEMATICA E INFORMATICA, UNIVERSITÀ DI UDINE, VIALE DELLE SCIENZE 206, 33100 UDINE, ITALY
 E-mail address: alberto.marcone@uniud.it
 URL: <http://users.dimi.uniud.it/~alberto.marcone/>